Mathematica Code for Tuple Imagery: Self-Similarity Without Complex Numbers

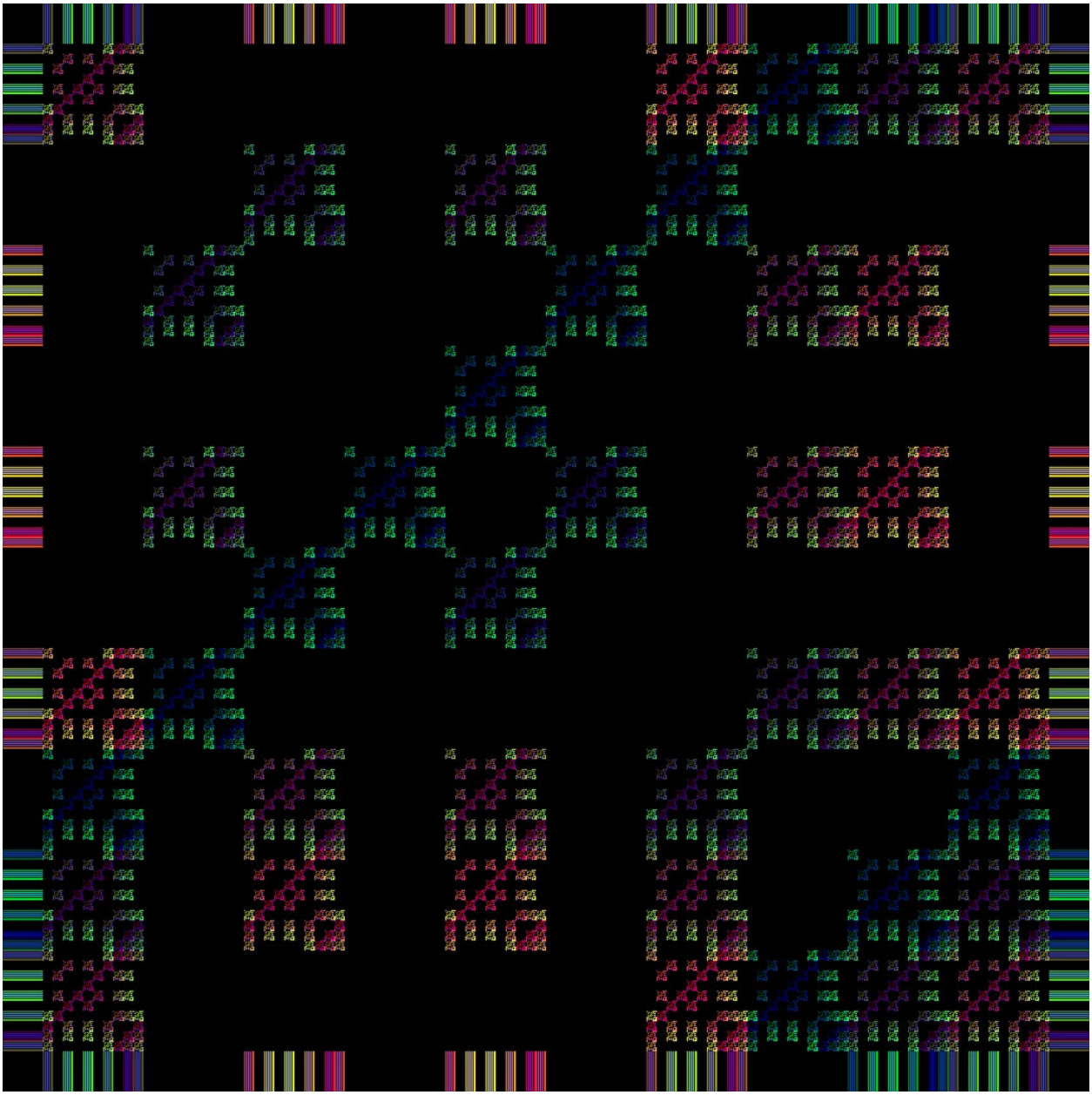
Douglas C. Youvan

doug@youvan.com

November 18, 2023

Here, we define the concepts of 'anti-color' and 'super-color' to be, respectively, lesser than and greater than the R, G, B values used in color imagery. These concepts are embodied in the following computer-generated graphics. One can think of the row of lines on the top and the right side of the image as comprising 1,080 different 'color guns'. Some of these guns 'fire' anti-color, and some shoot real color. Only the ‘real color’ guns are visible in this example. Anti-color guns are outside real color and they are arbitrarily assigned to the color black (K) to help highlight nearby real colors. Interesting intersections appear when rows and columns containing color and anti-color combine by simple addition of their R, G, B triples.

Note: 540 frame YouTube video of tuple graphics:
https://www.youtube.com/watch?v=X3hsezqM8go



Frame 240.

The ‘invisible’ anti-color and visible color can intersect to make a real color dimmer or extinguish it entirely. Real colors can add to make super-color which is also 'invisible' and encoded as black (K). The 'small-medium-large' repetition pattern is ‘fractal-like’ and

microscopic structures mimic macroscopic levels at several levels of magnification, due in part to the ordering of numbers within the list of tuples. According to *Mathematica's* implementation of the function "Tuple":

Tuples [*list*, *n*] generates a list of all possible *n*-tuples of elements from *list*.

The elements of *list* are treated as distinct, so that Tuples [*list*, *n*] for a list of length *k* gives output of length k^n.

The order of elements in Tuples [*list*, *n*] is based on the order of elements in *list*, so that Tuples[{a_1, … , a_k}, *n*] gives {{a_1, a_1, … , a_1}, { a_1, a_1, … , a_2}, … , {a_k, a_k, … , a_k}}.

```
For[loop = 1, loop ≤ 540, loop++,
 a = (Pi / 270) * loop;
 tupstr = {a, 2*a, 3*a, 4*a, 5*a, 6*a, 7*a, 8*a, 9*a, 10*a};
 sinval = Chop[N[Sin[tupstr]]];
 (*Chop takes small numbers to zero*)


 backcol =.; backrow =.; sum =.; gsum =.;

 backrow = Table[{x, y}, {x, 1, 1080}, {y, 1, 1080}];
backcol = Table[{x, y}, {x, 1, 1080}, {y, 1, 1080}];

backrow[[All, All]] = {0.0, 0.0, 0.0};
backcol[[All, All]] = {0.0, 0.0, 0.0};

 tup = Tuples[sinval, 3];



 For[j = 1, j ≤ 1000, j++, backrow[[All, j + 40]] = tup[[j]]];  (* 1080-1000 /2 = 40 Pad guns*)
For[i = 1, i ≤ 1000, i++, backcol[[i + 40, All]] = tup[[i]]];

 sum = backrow + backcol;


 For[i = 1, i ≤ 1080, i++, For[j = 1, j ≤ 1080, j++,
   If[((0 > sum[[i, j, 1]]) || (sum[[i, j, 1]] > 1) ||
       (0 > sum[[i, j, 2]]) || (sum[[i, j, 2]] > 1) ||
       (0 > sum[[i, j, 3]]) || (sum[[i, j, 3]] > 1)),
    (sum[[i, j, 1]] = sum[[i, j, 2]] = sum[[i, j, 3]] = 0)]]];


 gsum = Rasterize[Image[sum, ImageSize → {1080, 1080}], AspectRatio → 1];

 (* Export["C:\\Users\\3000\\Desktop\\Sin10Tuple5\\"<>ToString[loop]<>".tif", gsum,
   "TIFF",ImageSize→{1080,1080}] ; *)

 Export["C:\\Users\doug\OneDrive\Pictures\Tuples\\" <> ToString[loop] <> ".tif",
  gsum, "TIFF", ImageSize → {1080, 1080}] ;
```

```
For[loop = 1, loop ≤ 540, loop + +,□a = (Pi/270) ∗ loop; □tupstr
= {a, 2 ∗ a, 3 ∗ a, 4 ∗ a, 5 ∗ a, 6 ∗ a, 7 ∗ a, 8 ∗ a, 9 ∗ a, 10 ∗ a}; □sinval
= Chop[N[Sin[tupstr]]]; □(*Choptakessmallnumberstozero*)□□□□□backcol
=.; backrow =.; sum =.; gsum =.; □□backrow
= Table[{x, y}, {x, 1,1080}, {y, 1,1080}]; backcol
= Table[{x, y}, {x, 1,1080}, {y, 1,1080}]; backrow[[All, All]]
= {0.,0.,0.}; backcol[[All, All]] = {0.,0.,0.}; □□tup
= Tuples[sinval, 3]; □□□For[j = 1, j
≤ 1000, j + +, backrow[[All, j + 40]] = tup[[j]]]; (*1080 − 1000/2
= 40Padguns*)For[i = 1, i ≤ 1000, i + +, backcol[[i + 40, All]]
= tup[[i]]]; □□sum = backrow + backcol; □□□For[i = 1, i
≤ 1080, i + +, For[j = 1, j ≤ 1080, j + +,□If[((0
> sum[[i, j, 1]])||(sum[[i, j, 1]] > 1)||□(0
> sum[[i, j, 2]])||(sum[[i, j, 2]] > 1)||□(0
> sum[[i, j, 3]])||(sum[[i, j, 3]] > 1)),□(sum[[i, j, 1]] = sum[[i, j, 2]]
= sum[[i, j, 3]] = 0)]]]; □□□gsum
= Rasterize[Image[sum, ImageSize → {1080,1080}], AspectRatio
→ 1]; □□(*Export["C:\\Users\\3000\\Desktop\\Sin10Tuple5\\" <
> ToString[loop] <> ". tif", gsum, "TIFF", ImageSize
→ {1080,1080}]; *)□□Export["C:\\Users\doug\OneDrive\Pictures
\Tuples\\" <> ToString[loop] <> ". tif", gsum, "TIFF", ImageSize
→ {1080,1080}];
```